基于 DNN,RNN 和 CNN 的文本情感分类

张铭宇 [1]

Abstract

本实验的目的是实现三种文本分类算法，这里选择了三种具有代表性的文本分类算法: DNN,TextRNN 和 TextCNN, 测试的数据集是文本情感数据集, 有 MR 数据集 [2]和 SST-1 数据集 [3] , 这里通过比较三种算法在不同数据集上的结果来分析他们的适用场景.

1. 实验任务

本实验的主要任务是实现对情感分类数据集的分类，其中用到的数据集如下所示:

数据集	类别	训练集	验证集	测试集
MR	2	8000	1000	1000
SST-1	5	8544	1101	2210

表 1. 本实验使用的数据集: MR 为电影影评数据集，分为 pos 和 neg 两种，各有 10000 条; SST-1 为 MR 的扩充，具有五种情感类别，数据并不均衡

2. 文本预处理

文本预处理是自然语言处理的第一步，它可以减少文本的噪声，有利于提高分类模型的性能以及分类速度。由于情感分类数据集来自互联网的用户评论，其并没有进行文本的预处理，训练集中包含了大量的停用词，标点符号等。这些噪声没有提供对分类有用的特征通过去除这些内容可以提高分类速度与效果。

1. 去除标点符号: 由于标点符号往往并不能包含信息，所以通常第一步先将非英文字符去除或替换，使用 Python 中的字符串替换函数即可完成这一功能。
2. 去除标点符号: 由于标点符号往往并不能包含信息，所以通常第一步先将非英文字符去除或替换，使用 Python 中的字符串替换函数即可完成这一功能。
3. 大小写统一: 由于英文语法特定，会出现同一个单词不同大小写的情况，实验中通过扫描所有字符，将所有英文字符都转换为小写字符，即可完成大小写统一。
4. 去除停用词: 停用词是指在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据（或文本）之前或之后会自动过滤掉某些字或词，这些字或词即被称为 Stop Words（停用词），例如 a, the 等等。Python 的 sklearn 库提供了英语停用词列表，通过对语料库中的所有单词进行比对，即可去除这些无用的单词。
5. 去除低频词: 一些常见的出现频率很低的词，例如特定的地名，人名，这些词语很可能在词向量字典中不存在，而且它们会干扰分类，造成模型过拟合等等问题，所以通过统计数据集中的词频，去除低频词（出现次数 <2）即可将这些词尽可能去除。
6. 词向量化: 每一个单词是一个一维的向量，为了提高分类器的效果，这里将每个单词转换到相应的向量，本实验的词向量来自于预训练的 glove 词向量，其中每个单词向量的维度为 300，通过读入预训练的词向量文件，即可将每个单词映射到相应的向量，完成词向量化。这其中可能会存在未登录词的情况，由于未登录词所占单词表的比例很低，所里统一给他们赋值为一个随机的 300 维向量。由于 Tensorflow 提供了 embeding 层，所以这里只需要将词语替换为相应的序号，然后将序号对应的向量矩阵赋值给 embeding 层即可。由于句子长度是不固定的，但是网络输入大小却是固定的，所以需要设定一个输入大小 L，然后将长度超过 L 的句子进行截断，长度小于 L 的句子进行补齐。

经过以上步骤，分类的文本被转换为了一个矩阵，

矩阵的行是每个句子的

3. 实验数据集

MR. 电影评论数据集，每个评论为一个句子。评论类别包括正面/负面由于 MR 数据集没有划分好测试集验证集和训练集，所以这里按照分层抽样的方法，按照 0.8:0.1:0.1 的比例划分训练集，验证集和测试集 [2]。

SST-1. Stanford Sentiment Treebank 数据集，是 MR 数据集的拓展，每个评论有一个 [0,1] 之间的情感得分，数据集本身提供了 train/dev/test 的划分，本实验需要将连续的情感值离散化，分割为非常积极，正、中性、负、非常负五个情感类别 [3]。

实验在训练集上训练模型，然后通过验证集上的分类表现来调整超参数优化分类效果，然后最后在测试集上进行测试。

4. 损失函数

虽然本实验使用了三种分类算法，但是他们都是基于神经网络的分类算法，所以损失函数是一样，都采用了 cross entropy 来衡量训练时的分类损失，class 类的损失计算如式1所示：

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) \qquad (1)$$

5. 基于 DNN 的文本分类算法

DNN 就是最简单最原始的神经网络模型，又叫感知机模型（Perceptron）它由输入层，隐藏层和输出层组成。由于本实验是进行文本分类，所以需要在输入层前加入 embeding 层，将单词序列转换为向量模式，然后再进行向后传播。

5.1. 模型结构

本实验构建的模型如图4所示：由于 DNN 容易过拟合，所以本实验在隐藏层中加入了 dropout 层来降低过拟合，提高模型的泛化能力。

5.2. 超参数设置

由于本实验在两个数据集上做测试，所以选用了不同的超参数来得到最佳的分类的效果。

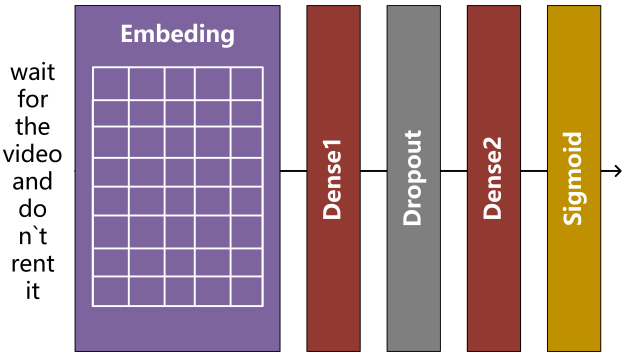

图 1. DNN 分类模型示意图

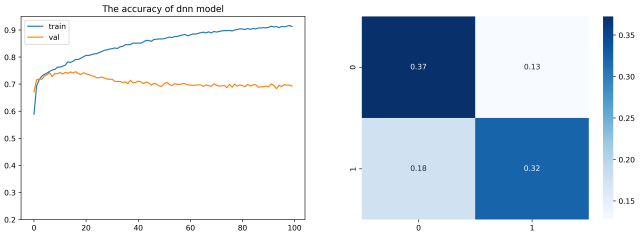

图 2. DNN 模型在 MR 数据集上的训练与测试

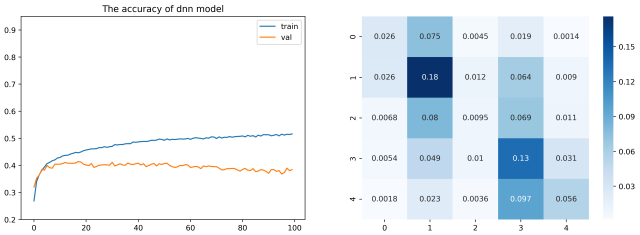

图 3. DNN 模型在 SST-1 数据集上的训练与测试

- MR: dropout rate (p) 为 0.5, mini-batch size 为 128，epoch 为 100，隐藏层单元数为 16，learning rate 为 0.001。
- SST-1: dropout rate (p) 为 0.5, mini-batch size 为 128，epoch 为 100，隐藏层单元数为 64，learning rate 为 0.001。

5.3. 实验结果

本实验的训练过程 loss 曲线以及在 MR 测试集上的表现如图2所示. 在 SST-1 测试集上的表现如图3所示

6. 基于 RNN 的文本分类算法

和传统的神经网络单一输入单一输出不同，循环神经网络可以输入一个序列信息，然后得到一个分类值或者另外一个序列。这种特性使得 RNN 可以考虑序

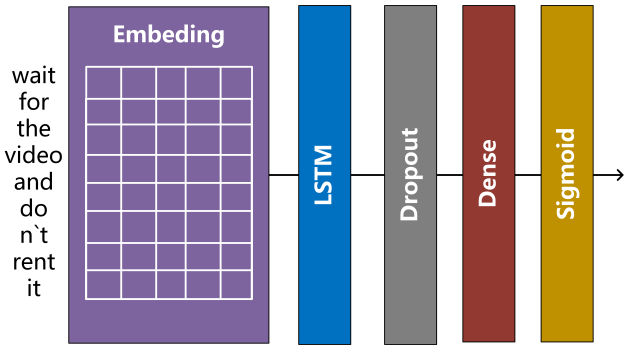

图 4. RNN 分类模型示意图

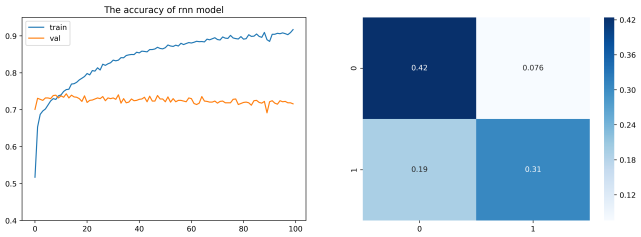

图 5. RNN 模型在 MR 数据集上的训练与测试

列信息的上下文信息，在机器翻译，词向量化等 NLP 问题上得到了广泛的应用。

本实验采用了 LSTM 作为文本分类的 RNN 单元，和 DNN 一样，首先对句子做词嵌入，然后将句子向量输入到 lstm 中，然后将 lstm 的输出再传入输出层全连接层中作为输出的预测值。

6.1. 模型结构

本实验构建的模型如图??所示：

6.2. 超参数设置

本实验中 RNN 的超参数设置如下：lstm 层数为 8，dropout rate (p) 为 0.5, mini-batch size 为 128，epoch 为 100，learning rate 为 0.001。

6.3. 实验结果

本实验的训练过程 loss 曲线以及在 MR 测试集上的表现如图5所示. 在 SST-1 测试集上的表现如图6所示

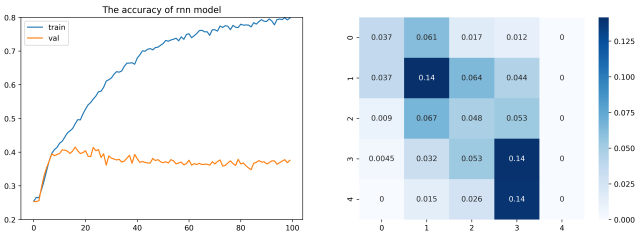

图 6. RNN 模型在 SST-1 数据集上的训练与测试

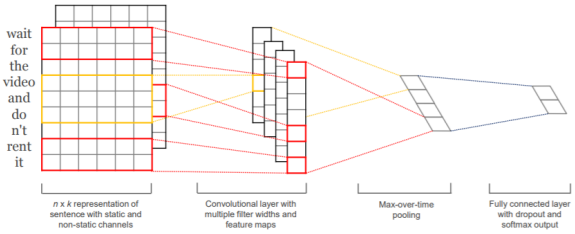

图 7. TextCNN 分类模型示意图

7. 实验结果

8. 基于 TextCNN 的文本分类算法

TextCNN 是由 Yoon Kim [1]提出的一种用来给句子进行分类的模型. 本实验实验的 cnn 模型如图7所示。TextCNN 利用多个不同 size 的 kernel 来提取句子中的关键信息，从而能够更好地捕捉局部相关性。和图像分类中的卷积层不同，由于文本是一维数据，因此在 TextCNN 卷积用的是一维卷积，因此需要通过设计不同大小的 filter 获取不同宽度的视野。

8.1. 超参数设置

由于本实验在两个数据集上做测试，所以选用了不同的超参数来得到最佳的分类的效果。本实验中 CNN 的 filter 大小为为 3，4，5，数目分别为 100，dropout rate (p) 为 0.5, mini-batch size 为 128，epoch 为 100，L2 惩罚系数为 3, learning rate 为 0.001。

8.2. 实验结果

本实验的训练过程 loss 曲线以及在 MR 测试集上的表现如图8所示. 在 SST-1 测试集上的表现如图9所示

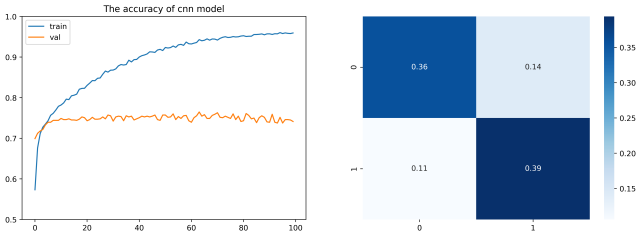

图 8. CNN 模型在 MR 数据集上的训练与测试

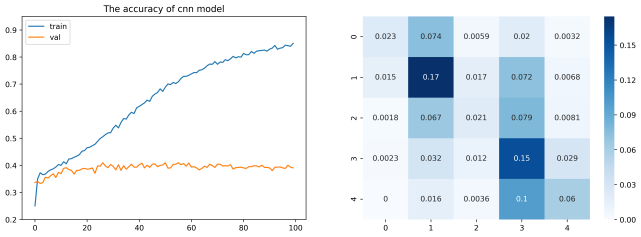

图 9. CNN 模型在 SST-1 数据集上的训练与测试

9. 结果分析

本实验三种模型的分类表现见表2 从表中可以看到，从 DNN 到 RNN 再到 CNN，模型的分类效果越来越好，而且 CNN 的表现要明显好于其他两种分类算法。

Model	MR	SST-1
DNN	0.6961	0.4023
RNN	0.7308	0.4258
CNN	0.7514	0.4339

表 2. DNN,RNN 和 CNN 的对比

参考文献

[1] Y. Kim. Convolutional neural networks for sentence classification. empirical methods in natural language processing, pages 1746–1751, 2014. 3

[2] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the ACL, 2005. 1, 2

[3] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Parsing With Compositional Vector Grammars. In EMNLP. 2013. 1, 2